AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A method that predicts a result produced by a section of code in order to support speculative program execution, the section of code including a plurality of program instructions, the method comprising:

executing the section of code within a program using a head thread, wherein executing the section of code produces the result;

before the head thread produces the result, generating a predicted result to be used in place of the result;

allowing a speculative thread to speculatively execute subsequent code within the program using the predicted result, wherein the subsequent code follows the section of code in an execution stream of the program~~, and wherein speculatively executing the subsequent code involves performing one of:~~

~~a speculative method invocation to speculatively execute the subsequent code,~~

~~a speculative function call to speculatively execute the subsequent code, and~~

~~a speculative procedure call to speculatively execute the subsequent code;~~

~~wherein the head thread and all speculative threads execute instructions from separate instruction caches~~; and

after the head thread finishes executing the section of code, determining if a difference between the predicted result and the result generated by the head thread affected execution of the speculative thread;

if the difference affected execution of the speculative thread, executing the subsequent code again using the result generated by the head thread; and

if the difference did not affect execution of the speculative thread, performing a join operation to merge state associated with the speculative thread with state associated with the head thread;

wherein during a write operation to a memory element by the head thread, the write operation involves:

performing the write operation to a primary version of the memory element,

checking status information associated with the memory element to determine if the memory element has been read by the speculative thread,

if the memory element has been read by the speculative thread, causing the speculative thread to roll back so that the speculative thread can read a result of the write operation, and

if the memory element has not been read by the speculative thread, performing the write operation to a space-time dimensioned version of the memory element if the space-time dimensioned version exists; and

wherein performing the join operation involves merging the space-time dimensioned version of the memory element into the primary version of the memory element and discarding the space-time dimensioned version of the memory element.

2. (Original) The method of claim 1, wherein executing the subsequent code again involves performing a rollback operation for the speculative thread to undo actions performed by the speculative thread.

3. (Original) The method of claim 1, wherein determining if the difference affected execution of the speculative thread involves determining if the speculative thread accessed the predicted result.

4. (Original) The method of claim 1, wherein determining if the difference affected execution of the speculative thread involves determining if the predicted result differs from the result generated by the head thread.

5. (Original) The method of claim 1, wherein generating the predicted result involves looking up a value based upon a program counter for the program.

6. (Original) The method of claim 5, wherein generating the predicted result involves additionally looking up the value based upon at least one previously generated value for the result.

7. (Original) The method of claim 5, wherein generating the predicted result involves performing a function on the value.

8. (Original) The method of claim 1, wherein executing the section of code involves performing one of:

a method invocation to execute the section of code;

a function call to execute the section of code; and

a procedure call to execute the section of code.

9. (Original) The method of claim 1, wherein the section of code is a body of a loop in the program, and the result is a loop carried dependency for the loop.

10-11 (Canceled).

12. (Currently amended) An apparatus that facilitates predicting a result produced by a section of code in order to support speculative program execution, the section of code including a plurality of program instructions, the apparatus comprising:

a head thread that is configured to execute the section of code within a program, wherein executing the section of code produces the result;

a prediction mechanism that is configured to generate a predicted result to be used in place of the result before the head thread produces the result;

a speculative thread that is configured to speculatively execute subsequent code within the program using the predicted result, wherein the subsequent code follows the section of code in an execution stream of the program~~, and wherein speculatively executing the subsequent code involves performing one of:~~

~~a speculative method invocation to speculatively execute the subsequent code,~~

~~a speculative function call to speculatively execute the subsequent code, and~~

~~a speculative procedure call to speculatively execute the subsequent code;~~

~~wherein the head thread and all speculative threads execute instructions from separate instruction caches~~; ~~and~~

a determination mechanism that is configured to determine if a difference between the predicted result and the result generated by the head thread affected execution of the speculative thread; and

a joining mechanism that is configured to merge state associated with the speculative thread with state associated with the head thread if the difference did not affect execution of the speculative thread, wherein the joining mechanism is configured to:

merge the space-time dimensioned version of the memory element into the primary version of the memory element, and to

discard the space-time dimensioned version of the memory element; and

a mechanism that performs write operations for the head thread, the mechanism being configured to:

perform a write operation to a primary version of a memory element,

check status information associated with the memory element to determine if the memory element has been read by the speculative thread,

cause the speculative thread to roll back so that the speculative thread can read a result of the write operation if the memory element has been read by the speculative thread, and

perform the write operation to a space-time dimensioned version of the memory element if the space-time dimensioned version exists and if the memory element has not been read by the speculative thread;

wherein if the difference affected execution of the speculative thread, the apparatus is configured to execute the subsequent code again using the result generated by the head thread.

13. (Original) The apparatus of claim 12, wherein while executing the subsequent code again, the apparatus is configured to perform a rollback operation for the speculative thread to undo actions performed by the speculative thread.

14. (Original) The apparatus of claim 12, wherein the determination mechanism is configured to determine if the speculative thread accessed the predicted result.

15. (Original) The apparatus of claim 12, wherein the determination mechanism is configured to determine if the predicted result differs from the result generated by the head thread.

16. (Original) The apparatus of claim 12, wherein the prediction mechanism is configured to generate the predicted result by looking up a value based upon a program counter for the program.

17. (Original) The apparatus of claim 16, wherein the prediction mechanism is configured to generate the predicted result by additionally looking up the value based upon at least one previously generated value for the result.

18. (Original) The apparatus of claim 16, wherein the prediction mechanism is configured to generate the predicted result by performing a function on the value.

19. (Original) The apparatus of claim 12, wherein the section of code includes one of, a method, a function and a procedure.

20. (Original) The apparatus of claim 12, wherein the section of code is a body of a loop in the program, and the result is a loop carried dependency for the loop.

21-22 (Canceled).

23. (Currently amended) A computer-readable storage medium storing instructions that when executed by a computer cause the computer to perform a method that predicts a result produced by a section of code in order to support speculative program execution, the section of code including a plurality of program instructions, the method comprising:

executing the section of code within a program using a head thread, wherein executing the section of code produces the result;

before the head thread produces the result, generating a predicted result to be used in place of the result;

allowing a speculative thread to speculatively execute subsequent code within the program using the predicted result, wherein the subsequent code follows the section of code in an execution stream of the program~~, and wherein speculatively executing the subsequent code involves performing one of:~~

~~a speculative method invocation to speculatively execute the subsequent code,~~

~~a speculative function call to speculatively execute the subsequent code, and~~

~~a speculative procedure call to speculatively execute the subsequent code;~~

~~wherein the head thread and all speculative threads execute instructions from separate instruction caches~~; and

after the head thread finishes executing the section of code, determining if a difference between the predicted result and the result generated by the head thread affected execution of the speculative thread;

if the difference affected execution of the speculative thread, executing the subsequent code again using the result generated by the head thread; and

if the difference did not affect execution of the speculative thread, performing a join operation to merge state associated with the speculative thread with state associated with the head thread;

wherein during a write operation to a memory element by the head thread, the write operation involves:

performing the write operation to a primary version of the memory element,

checking status information associated with the memory element to determine if the memory element has been read by the speculative thread,

if the memory element has been read by the speculative thread, causing the speculative thread to roll back so that the speculative thread can read a result of the write operation, and

if the memory element has not been read by the speculative thread, performing the write operation to a space-time dimensioned version of the memory element if the space-time dimensioned version exists; and

wherein performing the join operation involves merging the space-time dimensioned version of the memory element into the primary version of the memory element and discarding the space-time dimensioned version of the memory element.

24. (Original) The computer-readable storage medium of claim 23, wherein executing the subsequent code again involves performing a rollback operation for the speculative thread to undo actions performed by the speculative thread.